Spread and shrink: Point of interest discovery and coverage with mobile wireless sensors

Tahiry Razafindralambo, Milan Erdelj, Dimitrios Zorbas, Enrico Natalizio

Spread and shrink: Point of interest discovery and coverage with mobile wireless sensors

Tahiry Razafindralambo [a], Milan Erdelj [b], Dimitrios Zorbas [c,*], Enrico Natalizio [b]

[a] *Inria Lille - Nord Europe, France*

[b] *Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, France*

[c] *Université de La Rochelle, L3i, France*

A B S T R A C T

In this paper we tackle the problem of deploying mobile wireless sensors while maintaining connectivity with a sink throughout the deployment process. These mobile sensors should discover some points of interest (PoI) in an autonomous way and continuously report information from the observed events to the sink. Unlike previous works, we design an algorithm that uses only local information and local interactions with surrounding sensors. Moreover, unlike other approaches, our algorithm implements both the discovery and the coverage phase. In the discovery phase, the mobile sensors spread to discover new events all over the field and in the second phase, they shrink to concentrate only on the discovered events, named points of interest. We prove that connectivity is preserved during both phases and the spreading phase is terminated in a reasonable amount of time. Real experiments are conducted for small-scale scenarios that are used as a "proof of concept", while extensive simulations are performed for more complex scenarios to evaluate the algorithm performance. A comparison with an existing work which uses virtual forces has been made as well. The results show the capability of our algorithm to scale fast in both discovery, coverage and shrinking phases.

1. Introduction

Observation of the physical world is a key application in many civilian (environment), military (battlefield) and industrial (structural monitoring) domain. Wireless sensor networks made these operations easier, cheaper and more accurate. Easier because wireless sensors network can use self-configuration techniques. Cheaper due to the decreasing cost of electronics. Accurate by deploying a huge number of sensors and using network connections which can provide and spread results in some instances of time. A proper placement of wireless sensors over the area of interest is a critical job since it provides accuracy and information diffusion to the network. These two requirements are the main objectives of this paper.

1.1. Problem statement

In the context of wireless sensor network, an observation area can be reduced to only some interesting points called points of interest to reduce the complexity and the cost while maintaining the accuracy. The concept of monitoring certain points or strategic locations in the sensor field instead of the whole field area reduces the costs of the deployment by reducing the number of used sensors and helps improve the coverage performance by giving

the ability of increasing the number of covering sensors. However, discovering the points of interest needs a complete exploration of the target area.

The points of interest can be discovered thanks to mobile sensors. The discovery process implies the correct spreading of the mobile sensors all over the field of interest. Depending on the number of mobile sensors and the size of the area, some points of interest may not be discovered. However, it is important to guarantee a correct spreading of the sensors to maximize the discovered area. Once the points of interest discovered, only a proportion of the mobile sensors are useful and have to stay on the field since they cover the points of interest. It actually means that the sensors have to shrink to a smaller network.

The shrinking phase allows the useless sensors to go back to their starting point after the discovery phase. Since not all the sensors are covering a point of interest, they are not used to capture information on the field. Instead of leaving the sensors on the field, in which they are prone to damages, they could be brought back to their initial position to reduce the cost of the deployment. Some are sent over the discovered points of interest to increase the observation accuracy or used as data relays from the point of interest to the sink. Therefore, the shrinking phase is important since it can reduce the cost of deployment, increase the accuracy, and strengthen the connectivity between the nodes and the sink.

Connectivity is important throughout the process, since the discovery and coverage (with the same devices) can reduce the time needed to retrieve useful information from the point of interest. Indeed, the time is minimized between the transition from discovery to coverage if the mobile wireless network connectivity is maintained all along the deployment procedure. This is an advantage of our approach over other algorithms in the literature.

The coverage of the whole target area and the point of interest are well-known problems but separately investigated in the literature. Fan et al. [13] and Wang et al. [20] provide some rich surveys on the deployment of mobile wireless sensors to cover a given target area. Some optimal, yet centralized, strategies are described in the above-mentioned surveys. Moreover, some authors such as Razafindralambo et al. [17] have proposed some localized algorithms for the same task. The literature as described by Erdelj et al. [12] cites some important works regarding point of interest coverage. In their work, Erdelj et al. [12] assume that the points of interest are already discovered and their objective is, thus, to send some nodes to cover these points of interest. In their work, they provide an algorithm with guaranteed connectivity between the base station and each point of interest. To the best of our knowledge, combining the discovery part and the point of interest coverage part has not yet been deeply investigated in the literature. This paper fills this gap by providing an algorithm that combines discovery and coverage at the same time.

1.2. Our contribution

In this work, we propose a novel approach for adapting a network with mobile sensors to a set of points of interest. The approach is split in two phases; the spreading and shrinking process. The algorithm we propose is distributed in nature and runs on all the sensors using only the one hop information from their neighbors. The key characteristics of the approach are summarized as it follows:

- Localized algorithm for discovery and coverage process. The decisions taken by the sensors are based only on local state (single-hop neighbors).
- Spread or discovery process which preserves the connectivity all along the deployment procedure.
- The spread process follows a grid coverage pattern to expand.
- Shrink or coverage process which preserve the connectivity between the sensors that had discovered the point of interest and the data sink following the grid lines. Moreover for the non-covering sensors, the connectivity is still kept.
- A spread and shrink process that provides a permanent coverage of the point of interest once discovered.

A visual example of our approach can be seen in Fig. 1. This figure shows the time and space evolution of the sensors running the algorithm. The spreading process occurs from time 0 s to time 400 s. The shrinking process starts after the 400 s.

1.3. Related work

Our work tackles different operational issues related to point of interest coverage. Specifically, it implies exploration of the sensor field (discovery of the points of interest), monitoring of the points of interest and data gathering. Moreover, our work guarantees connectivity with the sink throughout all the phases of the approach. In this section, we analyze only the most fundamental works referring to these issues.

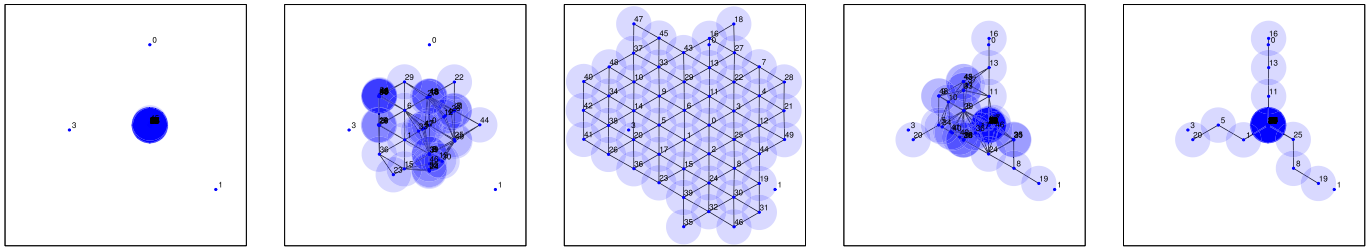
The coverage and monitoring of a point of interest or of an area of interest are subjects covered from both the ad-hoc, sensor and robotics community by using different approaches and by focusing on different aspects. The ad-hoc and sensor community consider devices such as sensors and actors, whose power as well as computational and execution capabilities are limited. The robotics community takes into account smarter and more powerful devices and assumes that communications do not have a basic importance in achieving the coverage of the area. They call this kind of approaches “formation control”. Since the topic has been extensively treated by both the communities in recent years, we will focus on the efforts produced by the ad-hoc and sensor community, which are more relevant to our work. For a survey on robotics formation control techniques please refer to the work of Wang et al. [18].

Younis et al. [24] and Wang et al. [20], authors survey and classify strategies and techniques for node placement and movement strategies for improving network coverage, respectively. They propose to classify works according to the targeted coverage. Specifically, full coverage aims at completely covering the field by geographically distributing sensors and actors on the entire area of interest in order to continuously monitor it.

Zou et al. [27] and Wang et al. [19] use mobility of the nodes in a more extensive way, in fact they propose target localization and sensor deployment, respectively, by using virtual forces. Our approach, also, considers a large use of mobile nodes as the last two cited works, but the main difference is in the algorithm that drives the nodes movements. We do not use virtual forces to avoid limiting the mobility of nodes only to the attraction–repulsion mechanisms and, also, to provide the grid coverage approach. Moreover, we redeploy the sensors in the shrink phase to focus only on the discovered points of interest and to reduce the number of sensors involved in the network.

Cheng et al. [8] study the maximum coverage problem in complex urban scenarios. The authors provide a geometry-based coverage strategy to handle the deployment problem over urban scenarios. This work is very interesting since it considers obstacles. However, the work does not consider the shrinking phase described in this paper. We will include the constraints raised in this paper regarding mobility pattern in our future work.

It is, also, important to notice the work of Bartolini et al. [1] which describes a hexagonal tilling deployment for a complete coverage of the area of interest. We use the same tilling in our work for the discovery (spreading) process since the hexagonal tilling guarantees at the same time network connectivity and optimal



(a) 0 s. (b) 100 s. (c) 400 s. (d) 530 s. (e) 630 s.

Fig. 1. Spread and shrink: At the beginning (time 0 s, (a)), when the spreading process starts, the sensors are located close to the data sink. The spreading process continues (time 100 s, (b)). At time 400 s, (c) the spreading process has ended up and the shrinking process starts. This shrinking process continues where the sensors come back to the data sink (time 530 s, (d)) until the final shape of the network (time 630 s, (e)).

coverage as shown by Brass [6] and Kershner [15]. However, the work of Bartolini et al. [1] differs from ours since our objective is both discovery and coverage. Moreover, our spreading technique is easier to implement it since all the sensors have the same role and we do not differentiate them by providing a slave and master status.

The work that is the closest to ours, in terms of objectives, is proposed by Erdelj et al. [10]. The proposed scheme uses the concept of delay tolerant networks to provide a controlled reachability as proposed by Whitbeck et al. [21]. On one hand, the main advantage of the scheme proposed by Erdelj et al. [10] is the minimized number of mobile sensors requisite in the network. On the other hand, the main problem of the scheme is the intermittent connectivity provided which strongly depends on the maximum and minimum speed of the mobile sensors. In this paper, we provide a point of interest discovery and coverage solution which provides a full-time connectivity all along the spread (discovery) and shrink (coverage) procedure.

Another type of deployment is the barrier coverage where sensors move to monitor a specific strip-shaped piece of area. The barrier coverage can be seen as the formation of a line of sensors between two points. It is related to our work since we provide the connectivity between the point of interest and the data sink by using sensors. However, our work differs substantially from the barrier coverage in terms of objectives, therefore we mention only Kumar et al. [16], which proposed the first theoretical study of this problem and Chen et al. [7] that proposes an interesting localized algorithm. These two works on barrier coverage provide some mechanisms to build a dense barrier of sensors in order to detect intrusions.

Finally, we report a work from the robotics community that considers exploration and coverage of the network proposed by Batalin et al. [2]. The exploration phase is performed by mobile robots and driven by a network of radio beacons which assists the robot(s) also during the coverage. On the contrary, our scheme does not consider any additional pre-deployed hardware for achieving the same objectives.

The rest of the paper is organized as follows: General assumptions, notations and definitions are presented in Section 2. Details of our algorithm are given in Section 3 with the description of the discovery and coverage phase. Section 4 analyzes the properties of the algorithm. In Section 5, we discuss the properties of the algorithm and our assumptions. Section 6 is devoted to the implementation and evaluation of our algorithm. Conclusions are drawn in Section 7.

2. Background, assumptions and definitions

In this section we provide the reader with the set of assumptions and definitions that will facilitate the explanation of our proposed deployment algorithm.

2.1. General assumptions

We assume that a set of mobile wireless sensors or mobile nodes $n_i, i = 1, .., N$ is placed in the field of interest with the goal of discovering some points of interest $p_i, i = 1, .., P$ that are supposed to be static. Once discovered, the sensors have to cover these points of interest. The unnecessary sensors, that is the sensors that are not covering a point of interest or that are not used to create a communication path between the point of interest and the data sink noted with Σ, have to move back to the data sink.

Without knowing their absolute position, we also assume that at the beginning the sensors are located close to the sink. Absolute geographical positioning mechanism is neither needed nor assumed, however, sensors should be able to deduce their relative positions among each other using mechanisms such as the one used by Bartolini et al. [1]. Dedicated sectorized distance sensors implemented in each of the mobile robots can be assumed and easily implemented. Using this kind of technique, the robot would be able to deduce the relative position towards its neighbors in the network.

A point of interest is discovered and covered if the distance between a sensor and the point of interest is lower than a given distance. This distance is called hereafter as sensing range and noted with r_s. We assume that each sensor has the same sensing range. In order to avoid sensing holes and to ensure an optimized coverage, we assume in our triangular tessellation that each sensor is a center of a hexagon and can have six neighbors at a maximum distance of $r_s\sqrt{3}$. Without loss of generality, we assume that the communication range r_c is at least equal to $r_s\sqrt{3}$, to ensure at the same time the coverage and the connectivity when the sensors are deployed following the triangular tessellation. Note that this assumption can be relaxed as studied by Yun et al. [25]. Fig. 2 illustrates this notation with six possible points that are called spots and are the possible locations of the neighboring sensors.

2.2. Model

A sensor or node n_i is characterized by its unique identifier $i, i \in \{1, .., N\}$, the sensing range (r_s) and the communication range (r_c). We assume without loss of generality that geographical coordinates noted (x_{n_i}, y_{n_i}) can be obtained for node n_i, for the data sink Σ (x_Σ, y_Σ), for each spot $s_j \in \mathcal{S}$ (x_{s_j}, y_{s_j}) and for the points of interest $p_k \in \mathcal{P}$ (x_{p_k}, y_{p_k}). It is important to notice here that the coordinates of a spot, are related to the coordinates of the sensors. In Fig. 2 the coordinates of s_0 are (x_{n_i}, y_{n_i}) and for s_1 they are $(x_{n_i} + x_{n_i}.\cos(\frac{\pi}{3}), y_{n_i} + y_{n_i}.\sin(\frac{\pi}{3}))$.

We model our sensor network as an evolving graph $G(\mathcal{V}, \mathcal{E})$ where the set or vertices $\mathcal{V}$ is the set of sensors and the set of edges $\mathcal{E}$ is the set of communication links. The radio transmission model is a unit disk graph model. An edge exists between a couple of nodes $(n_i, n_j) \in \mathcal{V}^2$ if $d(n_i, n_j) \leq r_c$ where $d()$ is the euclidean

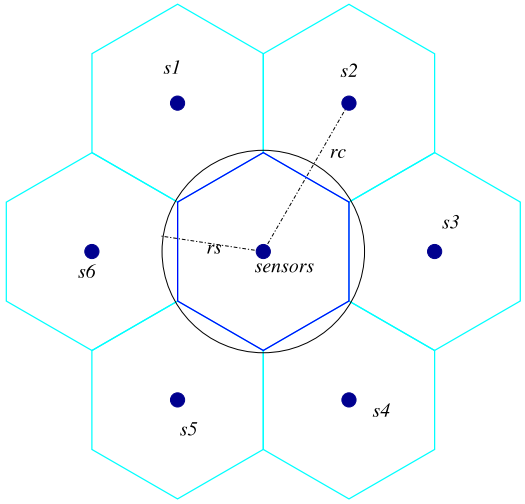


Fig. 2. Notation on the tessellation.

distance. The set of neighbors of node n_i is denoted by $\mathcal{N}^i$: $\mathcal{N}^i = \{n_k \mid n_k \in \mathcal{V} \wedge n_k \text{ is located on } s_j,\ j = 0, .., 6 \wedge n_k \neq n_i\}$. The node n_k is considered to be a neighbor of n_i if n_k is located at any spot s_j, $i = \{0, .., 6\}$ relatively to n_i. We say that n_i is connected to each of its neighbors $\in \mathcal{N}^i$.

We define the occupancy number ω_j of the spot $s_j, j = \{0, .., 6\}$ relatively to node n_i as the number of sensors located at the spot s_j. It is important to notice that at least one node is located at spot s_0, thus, $\omega_0 \geq 1$. In spot s_0 we can strictly order the sensors based on their unique identifier or the battery level or any combination of the metrics. For example if nodes n_i, n_j and n_k are located at the same spot and $i < j < k$, we say that n_i has priority $\rho_i = 1$, n_j has priority $\rho_j = 2$ and n_k has priority $\rho_j = 3$ on their actual spot. This occupancy number is used, for example in spot s_0, as a priority to address conflicting movement decisions to avoid loss of connectivity.

3. Deployment algorithm

This section presents our approach to target discovery and coverage with network connectivity preservation. We divide the description into two parts:

- *target discovery* (Section 3.1)—the goal is to discover the greatest area possible following a triangular tessellation while preserving the network connectivity,
- *target coverage* (Section 3.2)—the goal is to cover the discovered points of interest with the minimum number of sensors while preserving the network connectivity. Unused sensors come back to the base station or data sink location.

It is important to notice that our algorithm is localized. Each movement decision of the sensor is based on local information and the algorithm is executed at each node n_i except at the sink that is considered as a specific node. The algorithm does not need any kind of synchronization. We assume that the sensors use their wireless interface to periodically send a local broadcast message containing their unique id and their position.

3.1. Spreading

In the spreading phase, the network expands in order to discover as many points of interest as possible. The spreading phase can be divided into two different steps as it is described in the following.

D.A. **When to move**. First, the node has to decide whether it will/can move or not. We remind that node n_i is always located at spot s_0 since the spot enumeration is relative to each node. The node n_i decides to move based on the following condition:

❶ If the occupancy number of the actual spot (s_0) of node n_i is greater than or equal to λ (λ is a constant and $\lambda \geq 2$), that means that there is at least λ nodes (including n_i) on spot s_0 and the priority number of node n_i is greater than or equal to λ, that means that there is a least $\lambda - 1$ nodes co-located with n_i that have a higher priority. This condition can be summarized as follows: if $\omega_0 \geq \lambda$ and $\rho_i \geq \lambda$ then the node s_i can move.

It is important to notice here that the constant $\lambda \geq 2$ can be increased in order to provide $(\lambda - 1)$-coverage during the spreading phase.

At the end of this first phase, the node n_i knows if it can move or not.

D.B. **Where to move**. When a node is allowed to move, based on the previous condition, it has to choose its next position. Following the grid deployment, the possibility of each node is restricted to the six points (spots) of the triangle tessellation. The choice is based on a random drawing based on the following rules to create controlled behavior.

❷ Each spot s_i (relative to node n_i), $i \in [1, \ldots, 6]$, except s_0, is assigned a numeric value v_i, $i \in [1, \ldots, 6]$. This value depends on its occupancy. This is a random and uniform value drawn in the range: $[\omega_i \times C + C/4; (\omega_i + 1) \times C[$, where C is a constant. The next position of the node is the spot with the minimum value v_i.

❸ In order to make the network expand, the value v_i of a spot is set to infinity if this spot is closer to the data sink than the actual node's position and if this spot is occupied by "enough" nodes, where "enough" is related to λ. This condition can be summarized as follows: if $\{d(s_i, \Sigma) < d(n_i, \Sigma)$ and $\omega_i \geq \lambda - 1\}$ then $v_i = \infty$. If all the spots around a node n_i have infinite value, the node is not moving.

❹ In order to avoid a horizontal or a vertical only expansion, the value of a spot is reduced proportionally to the distance of the spot if the spot is closer to the sink. This condition can be summarized as follows: if $\{d(s_i, \Sigma) < d(n_i, \Sigma)$ and $\omega_i < \lambda - 1\}$ then $v_i = \frac{d(s_i,\Sigma)\times C}{4\times d(n_i,\Sigma)}$.

At the end of this second phase, the next position of the node n_i is the spot s_i with the minimum value v_i.

3.2. Shrinking

In order to ease the understanding, we define two status for a node. A *covering*-node is a node that is covering a point of interest and all other nodes are called *other*-node. During the spreading phase, some points of interest have already been discovered. In the current phase, the goal is twofold. First, we want to cover and keep the discovered points of interest covered. Second, we want all the *other*-node to come back to the sink while keeping the connectivity between the *covering*-node and the data sink. It is important to notice here that the process is fully distributed. This makes the algorithm very simple but makes it hard to provide an optimal point of interest coverage.

The shrinking phase strongly relies on the local construction of a connected dominating set (cds). A connected dominating set of a graph $G(\mathcal{V}, \mathcal{E})$ is a differentiation of nodes in $\mathcal{V}$. In a connected dominating set of a graph $G(\mathcal{V}, \mathcal{E})$ each node is either a dominant node or a neighbor of a dominant node (slave node). A local construction of a connected dominating set has been proposed by

Dai and Wu [9]. Hereafter, we describe the construction of the connected dominating set based on our assumptions.

We assume that each node n_i can be identified by a unique key κ_{n_i}. This key is also used to order the nodes. In our case κ_{n_i} is the distance between n_i and the data sink Σ and in case of tie the identifier is taken into account. Thus, we have $\kappa_{n_i} = \{d(n_i, \Sigma), i\}$. For two nodes n_i and n_j, $i \neq j$, we say that $\kappa_{n_i} > \kappa_{n_j}$ if $d(n_i, \Sigma) > d(n_j, \Sigma)$. When $d(n_i, \Sigma) = d(n_j, \Sigma)$, $\kappa_{n_i} > \kappa_{n_j}$ if $i > j$.

Construction of a connected dominating set. The construction of the connected dominating set follows these simple rules:

- A node n_i is dominant, if it has the lowest key value within its neighbors. This condition can be summarized as follows: if $\min_{(v \in \mathcal{N}^i)} \kappa_v = \kappa_{n_i}$ then n_i is dominant. If a node n_i is not dominant, that means that there exists a node $n_j \in \mathcal{N}^i$ with $\kappa_{n_i} > \kappa_{n_j}$. If κ_{n_j} is the minimum value within the neighbors of n_i, we say that n_j is the dominant node of n_i and we note $\delta_{n_i} = n_j$.
- A dominated node n_i locally broadcasts the identification of its dominant node δ_{n_i}. Each neighbor of n_i that receives this message and that is not δ_{n_i} evaluates its status without considering n_i.
- A node n_i is dominant if it has some neighbors that cannot be connected to δ_{n_i} by using neighbors with lower key value. This condition can be summarized as follows: if {for each $v \in \mathcal{N}^i$, if $(d(v, \delta_{n_i}) > r_c$ and $\nexists\{w \in \mathcal{N}^i | \kappa_w < \kappa_{n_i} \wedge d(w, \delta_{n_i}) < r_c\})$} then n_i is dominant.

Based on the above rules, the shrinking phase executed at node n_i is divided into two steps.

C.A. **When to move.** First, the node has to decide whether it will/can move or not. A node will move if:
- ❺ If the occupancy number of the actual spot (s_0) of node n_i is greater than or equal to λ. This condition can be summarized as follows: if $(\rho_i \geq \lambda)$ then n_i can move.
- ❻ If the occupancy number of the actual spot (s_0) of node n_i is lower to λ and it is not a *covering*-node. This condition can be summarized as follows: if $(\rho_i < \lambda$ and $d(n_i, p_k) > r_s, \ \forall p_k \in \mathcal{P})$ then n_i can move.
- ❼ If the occupancy number of the actual spot (s_0) of node n_i is lower to λ and it is not a dominant node. This condition can be summarized as follows: if $(\rho_i < \lambda$ and n_i is not dominant) then n_i can move.

C.B. **Where to move.** When a node is allowed to move, based on the previous conditions, it has to choose its next position.
- ❽ A node will move toward to the sink. This, also, corresponds to the position of δ_{n_i}.

It is important to notice here that the connected dominating set changes after every movement of a node so the status of a node changes from time to time.

4. Algorithm analysis

To ease the reading, we divide the analysis of our algorithm into the analysis of the spreading phase in Section 4.1 and the analysis of the shrinking phase in Section 4.2. In the following, we assume that there is no failure in the network, that is no node disappears during the whole process and all messages are sent. That is we assume no message loss. For the sake of simplicity we assume that the time can be discretized.

4.1. Spreading phase

Theorem 1. *Connectivity. During the spreading phase, if at time t_t the network is connected, at time t_T, $T > t$ the network is still connected whatever the movement decision of a node.*

Proof. Let us assume that at time t_t two nodes n_i and n_j are located on the same spot. Based on our algorithm, only one of these two nodes can move (Algorithm D.A.❶ with $\lambda = 2$) and leave its actual spot. Let us assume that node n_i has the ability to move because $\rho_i < \lambda$ and $\omega_0 = 2 = \lambda$. Since the next position of node n_i is on the spot s_k, $k \in [1, \ldots, 6]$ and that each spot is at most at distance r_c from s_0 thus after its movement, s_i is still connected to s_j. For any $\lambda > 2$ the same proof holds. □

Theorem 2. *Discovery. During the spreading phase, if at time t_t the discovered area is equal to $\mathcal{A}^t$, at time $t_T > t_t$ the discovered area is $\mathcal{A}^T \geq \mathcal{A}^t$.*

Proof. Let us assume that at time t_t two nodes n_i and n_j are neighbors and they are not located at the same spot. Let us assume that node n_i can move. Let us, also, assume that from node n_i point of view, n_j is located at the spot s_k. The covered area of the two nodes is $\mathcal{A}^t$. Let us assume that n_i can move to spot s_k at time t_T and that $\mathcal{A}^T < \mathcal{A}^t$. The following cases are possible:

- **case 1** There was only n_i located at its spot before moving. This is not possible since Algorithm D.A.❶ with $\lambda = 2$ shows that a node cannot move and leave a free spot.
- **case 2** Node n_i chooses the spot s_k. This means that $\omega_k < \lambda - 1$ from Algorithm D.B.❹ which is impossible since node s_j is located at spot s_k.
- **case 3** Node n_i chooses the spot s_k based on Algorithm D.B.❷ which means that ω_k is amongst the minimum occupancy. If ω_k is amongst the minimum occupancy, all the other spots are occupied by at least one node. Thus, $\mathcal{A}^T$ cannot be lower than $\mathcal{A}^t$ since we assume no failures and that **case 1** cannot occur.

The above impossible cases show that the discovered area could not be decreased. Therefore, the covered area is stable or increasing over time. □

Theorem 3. *Termination. The spreading phase terminates.*

Proof. When no nodes of the network are able to move, we say that the discovery phase reaches its end. Based on Algorithm D.B.❸ we know that a node will never come closer to the sink on a spot that is already covered. This property, combined with the above theorems guarantees that a visited spot will always be occupied (Theorem 1), that the covered area is expanding or stays stable (Theorem 2). Moreover, when a node reaches the border of the area it will not move (Algorithm D.B.❸). □

The above theorems, also, show that our algorithm maintains the connectivity all along the deployment procedure, that the covered area is maximized if the area has an infinite size and that the discovery algorithm will eventually terminate. We can rely on the termination of the discovery phase to start the covering or shrinking phase.

4.2. Shrinking phase

Lemma. *The connected dominating set produced by our algorithm is connected.*

Proof. The construction of our connected dominating set follows the one provided by Ingelrest et al. [14]. It is similar to the one described in [14] and the proof provided in [14] is also valid for our case. □

Theorem 4. *Coverage. If at time t_t, a point of interest p_k is discovered and covered, at time $t_T > t_t$, this point of interest is still covered.*

Proof. For the spreading phase, we rely on Theorem 1 which shows that a spot that is discovered is never left free again. Since the triangular tessellation is optimal in terms of coverage (based on the results of Brass [6] and Kershner [15]) and that the r_s is homogeneous for all nodes, then during the discovery phase a discovered point remains covered. For the shrinking phase, the condition described in Algorithm C.A.❻ says that if a node n_i is covering a point of interest and that the occupancy number $\omega_0 < \lambda$ (related to n_i) then the node cannot move. Therefore a node that is covering a point of interest will only leave its *covering*-node status it there exists another node on the same spot which is actually covering the same point of interest. □

Theorem 5. *Connectivity. During the shrinking phase, if at time t_t, the network is connected, at time $t_T > t_t$, the network is still connected.*

Proof. We show by Theorem 1 that at the end of the spreading phase, the network is still connected. Let us deal with the connectivity issue during the shrinking phase. Let us assume that at time t_t node n_i is moving. Based on Algorithm C.B.❽, the node will move to the position of its dominant node δ_{n_i}. Therefore after a movement, node n_i cannot be disconnected from the network since we assume no failures.

Now let us assume that node n_i disconnects node n_j from the network after its movement. It may happen in the following cases (note that in the following cases we assume that $\omega_0 < \lambda$ – related to n_i – otherwise n_j cannot be disconnected by the movement of n_i):

- **case 1** n_j is not a dominant node. Based on the construction of the connected dominating set if n_j is not dominant, it should be connected to a dominant node. In this case n_i might be the dominant node of n_j, which is impossible (based on Algorithm C.B.❼) since n_i is moving. If another node n_k was the dominant node of n_j, n_j should not be disconnected by the movement of n_i.
- **case 2** n_j is a dominant node. If n_i was not a dominant node, its movement should not disconnect the dominant node n_j. If it was the case, n_i should have been set as dominant based on the construction of the connected dominating set. If n_i was a dominant node, n_i is not able to move and, thus, not able to disconnect n_j.

Therefore if the network is connected at time t_t, the movement of any node cannot lead to disconnections if at time t_{t+1} it is still connected for any time $t_T > t_t$. □

Property 1. *During the shrinking phase, if no point of interest is discovered, all the nodes will come back and eventually reach the sink.*

Proof. Let us assume, without loss of generality, that the shrinking phase starts at the end of the spreading phase. Let us consider node n_i as the node with the highest $\kappa_{n_i} = \{d(n_i, \Sigma), i\}$ value at time t_t. Based on the connected dominating set construction, node n_i will be the first which will move towards its dominant node δ_{n_i}. By recursively applying the same mechanism with the node having the second highest κ value, all the nodes will eventually reach the sink. □

Property 2. *During the shrinking phase, if a point of interest is discovered and covered by only one node, this covering-node will eventually be a dominated node.*

Proof. Let us assume that node n_i is covering a point of interest and for the sake of simplicity we also assume that for all the future nodes that will pass onto the same spot as n_i, ρ_i is the minimum. We know from the property of Algorithm C.A.❻ that node n_i will not be able to move. If n_i is a dominant node, its dominated nodes will eventually move to its position. If n_i is needed to connect a dominant node n_j, n_j will eventually be a dominated node based on Property 1. Therefore node n_i will eventually be dominated. □

Theorem 6. *The shrinking phase eventually terminates.*

Proof. We assume that when a node reaches the sink, it does not move anymore. Therefore, based on Properties 2 and 1 and the conditions of movement described in Algorithm C.A.❺, C.A.❻ and C.A.❼, the algorithm will eventually terminate. Indeed, when *covering*-node is not moving, the dominant node of a *covering*-node is not moving either. By the same reasoning, the dominant node of a *covering*-node is not moving either and so on. □

Theorem 7. *Single path. If we assume that $\lambda = 2$ during the shrinking phase, a unique path will be created between each point of interest and the sink.*

Proof. From Property 2 we know that a *covering*-node will eventually be dominated. Let this *covering*-node be node n_c. It will have one and only one dominant node. Let us assume that n_i is the dominant node of n_c ($c \neq i$). Property 2 will eventually apply recursively to n_i and to the dominant node of n_i. Therefore there will be a path of dominant nodes between n_c and Σ. Since the nodes will have only one and only one dominant node (based on the value of κ), this path will be unique. □

Remark. It is important to notice that some nodes may be common for different points of interest. However, Theorem 7 shows that there could be no more than one path starting from a point of interest. It is, also, important to notice that if a *covering*-node is part of a path generated by another point of interest, the paths will be merged starting from this *covering*-node.

Theorem 8. *Shortest path. During the shrinking phase, the path created between a covering-node and the sink is the shortest path on the grid.*

Proof. Let us assume that the path created between the *covering*-node and the sink is not the shortest path on the grid. That means that there exists a node n_j, neighbor of n_i in the same path, which is closer to the sink that it is at the same time the dominant node chosen by n_i. That is, $d(\delta_i, \Sigma) > d(n_j, \Sigma)$ where $n_j \in \mathcal{N}^i$. This is impossible since in our connected dominating set construction n_i should have chosen n_j. Therefore, when the nodes are positioned on the grid, the path created by our algorithm during the shrinking phase is the shortest path. □

Remark. Theorem 8 shows that the path created is the shortest path in terms of distance. This path is also the shortest in numbers of hops when the nodes are placed on the grid. It is important to notice here that Theorem 8 is valid for each individual *covering*-node.

5. Discussion

In this section, we discuss the properties, the assumptions and the behavior of our algorithm. We focus on the strengths and weaknesses of our proposal and we give some rational regarding our implementation.

- On the use of grid property for connectivity during the spreading phase. We have chosen to use the grid to maintain the connectivity during the spreading phase instead of other means such as the property of the Relative Neighbourhood Graph such as described by Razafindralambo et al. [17] since the grid approach is easier to implement on real deployment and do not need position information such as the Relative Neighbourhood Graph.
- If two nodes cover the same point of interest, there can be two different paths from the two nodes towards the sink during

the shrinking phase since each *covering*-node can have its own dominant node. This behavior can be omitted at the cost of message exchange between the *covering*-nodes.

- We use a connected dominating set to maintain the connectivity during the shrinking phase and not the grid such as in the spreading phase or any graph reducing technique such as the one described by Razafindralambo et al. [17]. The implementation choice is based on the simplicity of the connected dominating set compared to other techniques. Indeed, using the grid to maintain the connectivity would recursively need each *connecting*-node to know to which node it should be connected to in order to be allowed to move. In practice, this behavior may at some point break the connectivity before restoring it again.
- In the analysis of our algorithm we did not consider the presence of obstacles to ease the analysis. However, our algorithm can easily handle them. The important parameter to take into account is κ_{n_i}. In order to handle obstacle κ_{n_i} should not be the euclidean distance but the number of hops to the sink Σ, in combination with the id of the node. κ_{n_i} should simply incremented them after each movement of a node. This is implemented and used in our simulation results.
- For the sake of analysis, we assume that the two phases (spreading and shrinking) are separate and successive. However, each sensor can run each phase independently. A sensor n_i can be running the spreading phase and a node $n_j, i \neq j$ can be running the shrinking phase at the same time. In our implementation, once a node cannot move for a given amount of time (proportional to its distance from the sink), it can switch to the shrinking phase without considering the state of the other nodes. The optimal waiting time could be computed but we leave this as a point of future work.
- Comparison with work in the literature is difficult since none of the work in the literature have the same objectives (such as the work of Chen et al. [7] and the work of Zou et al. [27]), assumptions (such as the work of Bartolini et al. [1]) and constraints (such as the work of Erdelj et al. [12]) as ours. Therefore, we do not provide any comparison with the work in the literature since it would be unfair due to different objectives and assumptions. However, for the sake of comparison, we mixed some techniques described in the literature and came up with a scheme based on virtual forces with the same objectives, assumptions and constraints. Specifically, the work of Zou et al. [27] was used with some connectivity enhancements of the work of [17]. More details are given in Section 6.2.
- The use of grid assumes that covering and sensing follow a regular disk pattern. However, in reality, it is not the case. The two works of Boukerche et al. [3,5], study the impact of irregular covering range. The works presented in [3,5] could be adapted to our deployment scheme to provide a mobile spread and shrink deployment with different sensing range and communication. For example, the algorithm could dynamically adapt the grid size and share this information to neighboring nodes or by allowing some intermediate nodes to stop in between spots to provide coverage and connectivity.
- Energy is an important issue in wireless sensor networks. This issue is even more important when sensors are mobile. The energy consumption of our algorithm is left for future work. However, we have preliminary results on energy model as described in a [26]. We will combine our work with the results described in [4] to introduce an accurate energy consumption model in our algorithm.
- Depending on the value of λ, multiple nodes can stay at a given spot. This can create robustness and avoid network disconnection in case of failure of a node. In our future work, we will investigate the effect of λ and will we try to enhance our algorithm by providing a disjoint multipath scheme for the shrinking phase.

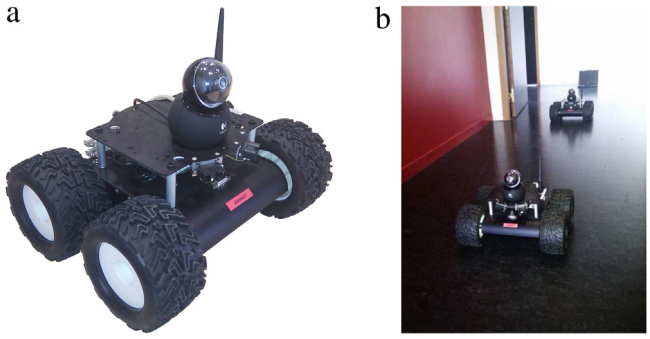


Fig. 3. Wifibot mobile robot (a) and a set of robots in an experimental scenario (b).

6. Implementation and evaluation

6.1. Robotic platform

We have implemented our algorithm on Wifibots mobile robots [22] (Fig. 3) by using the middleware provided by Erdelj et al. [11]. Our choice for the localization technique is the dead-reckoning localization based on the output from motor encoders (odometer). This simple localization method is widely used and is already implemented in the middleware. The Wifibot has a native IEEE802.11 network interface used for communication. We worked in an indoor environment and therefore we artificially modified the communication range and sensing range of the robots. We used this small scale implementation as a proof of concept.

Fig. 4 shows a simple example of the spread and shrink algorithm implemented on the Wifibot platform. In this figure, we assume that a point of interest is discovered when an obstacle is discovered by the robot. In our case an obstacle is the represented by the infrared detection of a wall. Fig. 4(a) shows the initial position of the robots. It is important to notice here that this initial position is capital for the algorithm since the angle of the tessellation is determined according to this initial position. This lack of robustness in the experimentation is due to the localization method we used. Fig. 4(b) and (c) show the spreading phase and Fig. 4(d) and (e) show the shrinking phase. In Fig. 4(e), we can see on the left of the picture the robot that has discovered a point of interest and therefore does not move back toward the data sink. It is important to notice that in this experiment, each robot goes to a different direction which was not always the case. Moreover, in the messages they exchange, the robots provide a relative position which is related to their initial one. Due to the lack of space in the experimentation facility and the limited number of robots, we could not run more experiments showing the phases with multiple hops. However, this example shows the feasibility of our algorithm.

6.2. Simulations

This section shows some performance evaluation results of our algorithm. Simulations were performed using WSNet [23]. We chose the WSNet simulator since the code written is the simulator and the one used in the robots are very similar and only few changes were needed. In the simulations, we set the communication range to be equal to the sensing range but this assumption can be easily modified without affecting the behavior of the deployment. Our algorithm is called "SaS" for Spread and Shrink in the following text.

In order to provide a benchmark to compare our algorithm, we use different results from the literature. We implement the virtual for calculation described in [27]. A basic comparison with this algorithm is unfair since it does not have any restriction about

Fig. 4. Deployment of six Wifibot. In (a) shows the initial position of the robots. The spreading phase takes place in (b) and (c). The shrinking phase are represented in (d) and (e).

Table 1
Summary of the simulation parameters.

Field size	500 m × 500 m starting at (250 m × 250 m)
Sensing range	20 m
Communication range	20 m (measured on wifibot)
Communication protocol	IEEE802.11 AdHoc mode (wifibot interface)
Hello Message	1 per second (measured on wifibot)
Decision period	30 s + random time
Maximal speed	0.9 m/s (measured on wifibot [26])

mobility and does not implement the shrinking phase. We have modified the results of [27] and add the connectivity preservation mechanisms described in [17] in order to keep connectivity all along the deployment procedure during the spreading phase. We have also implemented the shrinking phase by allowing specific nodes to shrink based on a localized connected dominating set election mechanism as described in [14]. The benchmark algorithm is called "VIR" for Virtual Forces in the following text.

The same simulation parameters are used to have a fair comparison. These simulation parameters are given in Table 1. It is also important to notice that for the sake of fairness and evaluation, we limited the simulation time to 2000 s for both algorithms. This is especially important for VIR since there is no other way to decide the end of each phase. We assumed for both algorithms that the spreading phase lasts 1000 s and the shrinking phase starts after 1000 s until 2000 s. We provide an evaluation of SaS deployment time in Section 6.5.

In order to ease the comprehension of the simulations, we use a fixed number of point of interest with always the same position. We define eight points of interest. The points of interest are placed at positions {(0, 85); (0, −85); (85, 0); (−85, 0); (75, 75); (75, −75); (−75, 75); (−75, −75)}, assuming that the sink is placed at coordinate (0, 0). Fig. 5 shows the configuration of the points of interest and the position of the sink. This configuration represents a point of interest distribution that allows us to evaluate the discovery and coverage phase.

It is important to notice that the results provided in this section are related to the assumptions we have made regarding the robotic platform we use. This section is meant to enrich the real deployment result with higher number of nodes, and points of interest.

6.3. Spreading phase evaluation

In this section, we provide some simulation evaluation of the spreading/discovery phase. We evaluate the number of discovered points of interest, related to the number of nodes and the time to discover the points of interest.

Fig. 6 shows an example of the discovery phase of SaS. We can see from this figure that despite the fact that only three points of interest are discovered within the predefined amount of time, the node distribution is uniform and no coverage holes appear. Connectivity was 100% guaranteed during the phase.

Fig. 7 shows an example of the discovery phase of VIR. We can observe that VIR discovers two points of interest more than SaS but the node distribution is not uniform and coverage holes appear in the network. Connectivity was, also, preserved by VIR.

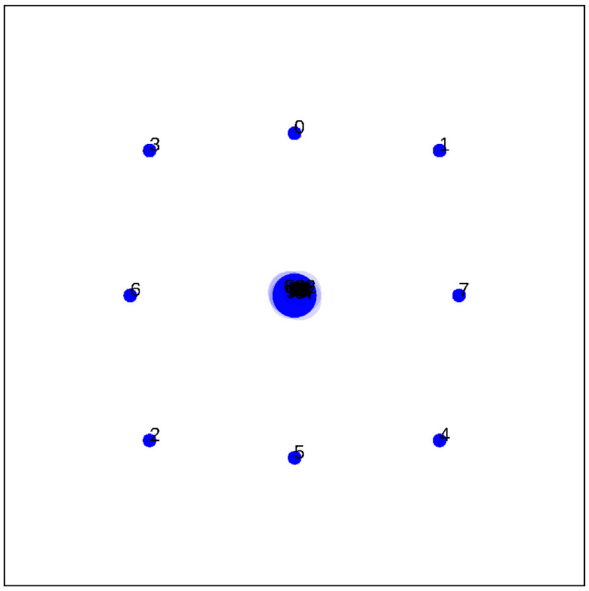


Fig. 5. Simulation configuration with the position of the points of interest.

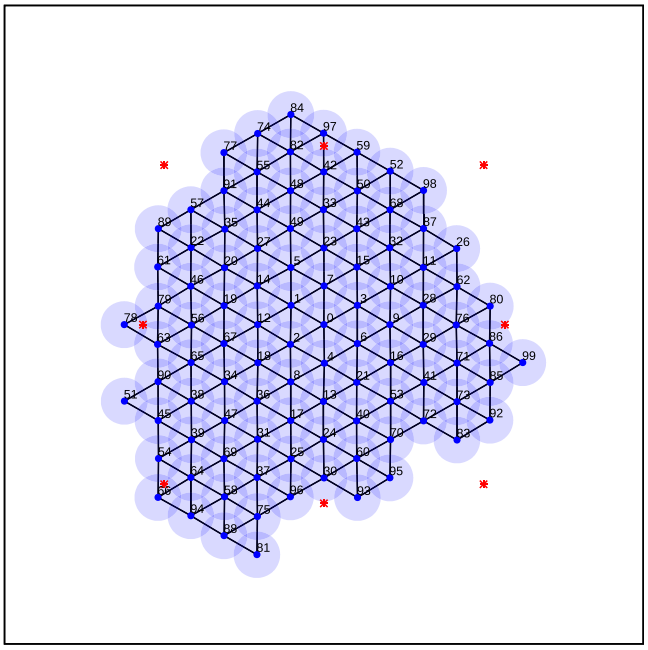


Fig. 6. Discovery phase with 100 nodes. Three points of interest are discovered by SaS.

Fig. 8 shows the number of discovered points of interest in relation with the number of mobile wireless sensors used. We can see from the figure that the discovery of all the point of interest depends on the number of nodes for both SaS and VIR. Since the movement decisions are random, the coverage is not always compact nor evenly distributed around the sink. This randomness

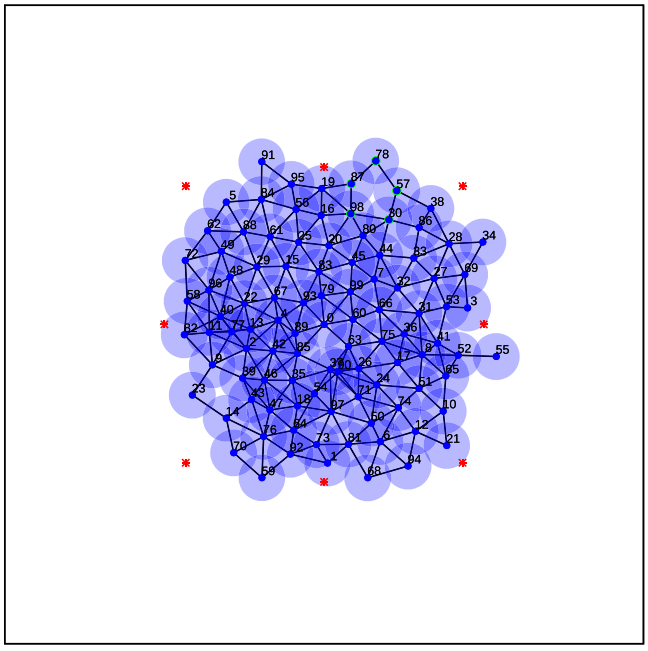


Fig. 7. Discovery phase with 100 nodes. Five points of interest are discovered by VIR.

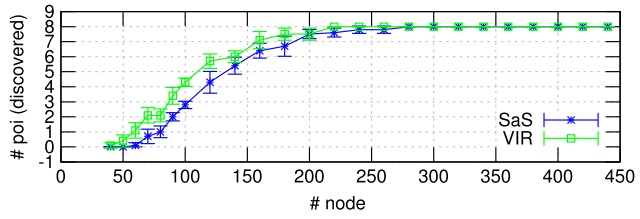


Fig. 8. Number of discovered point of interest depending on the number of nodes. 99% confidence interval is given.

shows that with very few nodes (60) it is possible to discover a point of interest. It is important to notice that, in the case of an even distribution of node around the sink it is not possible with 60 nodes to discover any poi given the configuration described above.

Fig. 8 also shows that with 280 nodes and above, all the points of interest are always discovered during all the simulations for both SaS and VIR. This behavior shows that the randomness of the algorithm is controlled and lead to a deployment that has statistical certainties. The simulation with 280 nodes was run 1200 times and for each of the occurrence, all the points of interest were discovered for both SaS and VIR.

We can point our from Fig. 8 that VIR has better performance than SaS regarding the point of interest discovery process. This is mainly due to the fact that VIR is not constraint by the grid positioning. Therefore it can discover more surface than SaS as pointed out in [17].

Figs. 9 and 10 give some examples of the discovery evolution of the points of interest for SaS and VIR respectively.

For SaS, we can see from Fig. 9 that the discovery speed is not strictly related to the number of nodes. This discovery speed is related to the decision period which is the same for all the simulation configurations. This discovery is faster with higher number of nodes since at each time the probability of having a node moving is higher. It is important to notice that the point of interest discovery is not related to the whole discovery phase duration.

For VIR, we can see from Fig. 10 that the discovery process takes more time. This behavior is due to the virtual forces process deployment. Since each movement is a combination of repealing

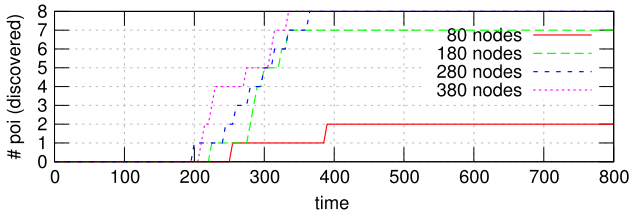


Fig. 9. Evolution of the discovery process for SaS. The figure plots some examples of the number of discovered points of interest depending on time with different number of nodes.

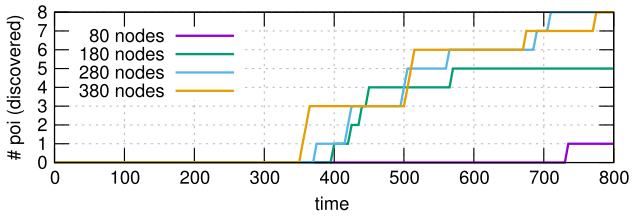


Fig. 10. Evolution of the discovery process for VIR. The figure plots some examples of the number of discovered points of interest depending on time with different number of nodes.

and attracting forces, the movement of a node can be small which increases the deployment time and, thus, the discovery time.

6.4. Shrinking phase evaluation

In this section we provide results regarding the shrinking/coverage phase. We specifically focus on the number of nodes used for the covering phase including the number of *covering*-nodes and *connecting*-nodes.

Figs. 11 and 12 show an example of the result of the coverage phase for both SaS and VIR. We can see from this figure that the discovered points of interest are covered and a communication path between the *covering*-nodes and the sink is created using *connecting*-nodes. In these simulations, we consider 100 nodes with 3 discovered points of interest for SaS and 5 for VIR.

Fig. 11 shows that there are no multiple paths between the *covering*-node and the sink. This result is due to the grid deployment configuration which avoids two nodes to be selected as dominant.

In Fig. 12 we can see that multiple paths can be created between the *covering*-node and the sink. While this may be seen as redundancy, we can see that the number of *connecting*-nodes is higher compared to SaS and that multiple paths are not often obtained.

Figs. 13 and 14 plot the number of used nodes at the end of the simulation time for both SaS and VIR. The number of nodes can be divided in two categories: *covering*-node and *connecting*-node.

For SaS, in Fig. 13 we can see that once there are enough nodes to discover all the points of interest, the number of used nodes is stable which is an expected behavior since the positions of the points of interest in our simulation are always the same. The increasing trends after 200 nodes deployed are due to the randomness of the deployment (we plot an average here) and, also, due to the fact that a point of interest can be covered by more than one *covering*-node. The behavior of our algorithm regarding the number of used nodes shows some kind of auto-stabilization property. Indeed, whatever the configuration at the end of the spreading phase if the same set of points of interest is discovered, the resulting number of used nodes will be roughly the same.

Regarding VIR, in Fig. 14, we can see that the number *covering*-nodes is stable but the *connecting*-nodes is increasing when the number of node increases. This behavior is due to the virtual force

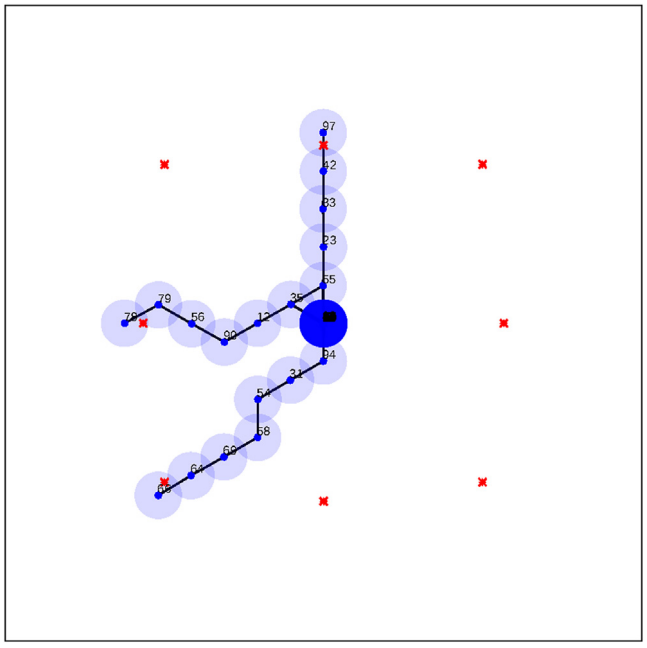


Fig. 11. Coverage phase with 100 nodes for SaS. Three points of interest are discovered, covered and connected with a path to the sink.

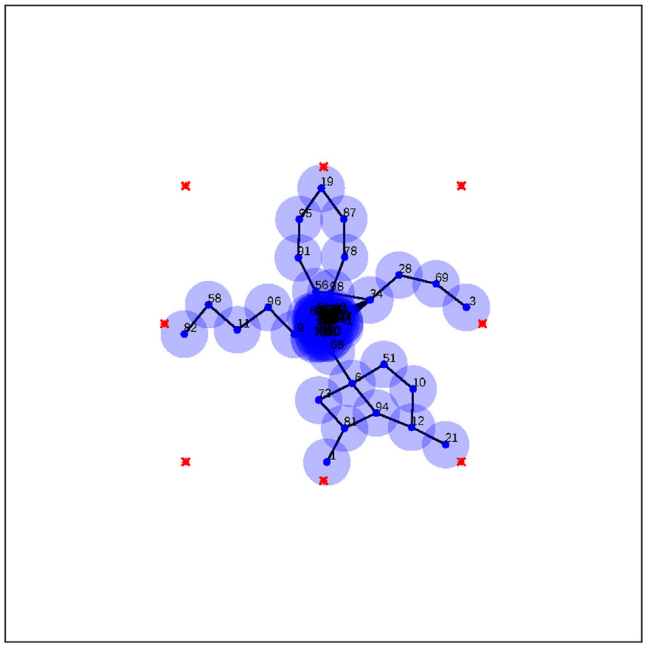


Fig. 12. Coverage phase with 100 nodes for VIR. Five points of interest are discovered, covered and connected with a path to the sink.

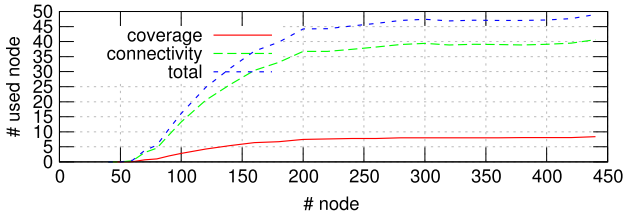


Fig. 13. SaS: Number of used nodes at the end of the shrinking phase depending on the number of initially deployed nodes. This graph plots the number of *covering*-node, the number of *connecting*-node and the total number of used nodes (sum of *covering*-node and *connecting*-node). For the sake of clarity, the confidence intervals are not plotted.

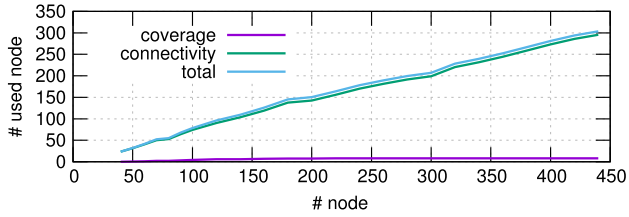


Fig. 14. VIR: Number of used nodes at the end of the shrinking phase depending on the number of initially deployed nodes. This graph plots the number of *covering*-node, the number of *connecting*-node and the total number of used nodes (sum of *covering*-node and *connecting*-node). For the sake of clarity, the confidence intervals are not plotted.

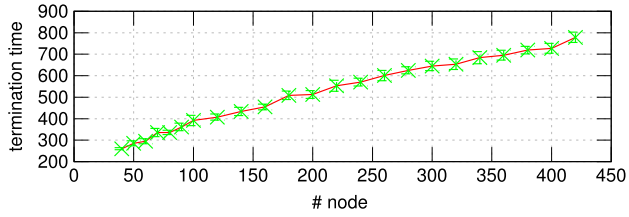


Fig. 15. Duration of the spreading phase depending on the number of node.

process. Indeed, since we add a connectivity constraint to the virtual force deployment, it reduces the movement range of each node to maintain this connectivity. This, therefore, reduces the shrinking phase evolution. Therefore after the simulation time of 2000 s, the shrinking phase may not be finished. It is important to notice that the shrinking phase is much longer that the spreading phase since we can assume that the distance between nodes is close to the communication range. This distance increases the value of the attractive forces among nodes and, thus, reduces the effect of the repulsive ones which should drive the nodes toward the sink.

6.5. Time deployment evaluation

In this subsection, we evaluate the duration of each phase of our algorithm. It is important to notice that we do not provide any comparison with the VIR algorithm since the VIR algorithm does not have a specific way to determine if the algorithm has reached its termination point.

Fig. 15 provides the duration of the discovery phase. This figure plots the duration of the discovery phase varying the number of nodes. This time is evaluated based on the definition of termination given in Theorem 3. We can see that the discovery phase time is linearly increasing the number of nodes. This linear behavior shows the scalability of our algorithm. It is important to notice that the only dependency on sensor decision is local and when sensors are more that one-hop apart, they can move at the same time and their decisions are independent. This simultaneity allows a fast deployment of the network.

Fig. 16 plots the duration of the shrinking phase varying the number of nodes. We can see that the shrinking phase duration is, also, linear with the number of nodes for the same reason we explained before about the spreading phase. It is important to notice that the shrinking phase of our algorithms is faster than the spreading phase. Indeed, *covering*-nodes and *connecting*-nodes are not moving which reduce the shrinking time since less nodes are considered.

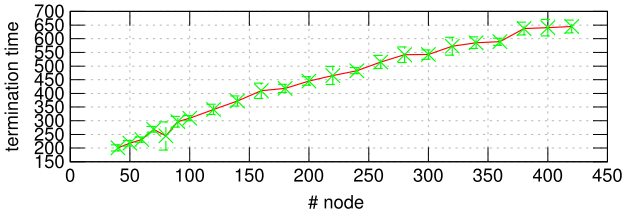


Fig. 16. Duration of the shrinking phase depending on the number of node.

7. Conclusion

In this work, we presented a distributed approach for multiple target discovery and coverage with mobile sensors. Target locations are unknown in the beginning of the deployment and, therefore, we introduced two separate phases of deployment: target discovery and coverage or spreading and shrinking. The algorithm relies on simple yet powerful interaction between nodes to get an efficient and robust discovery phase based on a grid deployment and the use of connected dominating set for the coverage phase. By combining these two phases we provide a complete and original solution for the discovery and coverage of unknown points of interest location in a field. Moreover, our algorithm ensures some properties of the network graph such as connectivity, termination, etc.

The performance of our algorithm is assessed using extensive simulation and the algorithm has, also, been implemented in a small scale testbed using robots to show the feasibility of our approach. Comparison with an algorithm which uses virtual forces to move the nodes has been made. The next step of this work is to consider the energy consumption in the local decisions and run some more experiments. We, also, plan to introduce network traffic and use the robots capability of movement as a way to improve quality of service in the network.

Acknowledgment

This work was partially supported by a grant from CPER Nord-Pas-de-Calais/FEDER DATA.

References

[1] N. Bartolini, A. Massini, S. Silvestri, P&p: an asynchronous and distributed protocol for mobile sensor deployment, Wirel. Netw. 18 (4) (2012) 381–399.
[2] M.A. Batalin, G.S. Sukhatme, Coverage, exploration and deployment by a mobile robot and communication network, in: Proceedings of the International Workshop on Information Processing in Sensor Networks, 2003, pp. 376–391.
[3] A. Boukerche, X. Fei, A coverage-preserving scheme for wireless sensor network with irregular sensing range, Ad Hoc Networks 5 (8) (2007) 1303–1316.
[4] A. Boukerche, X. Fei, R.B. Araujo, An energy aware coverage-preserving scheme for wireless sensor networks, in: Proceedings of the 2Nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN'05, ACM, New York, NY, USA, 2005, pp. 205–213.
[5] A. Boukerche, X. Fei, R.B. Araujo, An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange, Comput. Commun. 30 (14–15) (2007) 2708–2720. network Coverage and Routing Schemes for Wireless Sensor Networks.
[6] P. Brass, Bounds on coverage and target detection capabilities for models of networks of mobile sensors, TOSN 3 (2) (2007) 9.
[7] A. Chen, S. Kumar, T.H. Lai, Designing localized algorithms for barrier coverage, in: MobiCom'07: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, ACM, New York, NY, USA, 2007, pp. 63–74.
[8] H. Cheng, X. Fei, A. Boukerche, A. Mammeri, M. Almulla, A geometry-based coverage strategy over urban vanets, in: Proceedings of the 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, PE-WASUN'13, ACM, New York, NY, USA, 2013, pp. 121–128.
[9] F. Dai, J. Wu, Distributed dominant pruning in ad hoc networks, in: IEEE International Conference on Communications, 2003. ICC '03, Vol. 1, 2003, pp. 353–357.
[10] M. Erdelj, V. Loscrì, E. Natalizio, T. Razafindralambo, Multiple point of interest discovery and coverage with mobile wireless sensors, Ad Hoc Networks 11 (8) (2013) 2288–2300.
[11] M. Erdelj, T. Razafindralambo, Design and implementation of architecture for multi-robot cooperation in the context of wsn, in: PE-WASUN, 2013, pp. 33–40.
[12] M. Erdelj, T. Razafindralambo, D. Simplot-Ryl, Covering points of interest with mobile sensors, IEEE Trans. Parallel Distrib. Syst. 24 (1) (2013) 32–43.
[13] G. Fan, S. Jin, Coverage problem in wireless sensor network: A survey, J. Netw. 5 (2010) 1033–1040.
[14] F. Ingelrest, D. Simplot-Ryl, I. Stojmenovic, Smaller Connected Dominating Sets in Ad Hoc and Sensor Networks based on Coverage by Two-Hop Neighbors, Rapport de recherche RT-0304, INRIA, 2005, URL http://hal.inria.fr/inria-00069876.
[15] R. Kershner, The number of circles covering a set, Amer. J. Math. 61 (3) (1939) 665–671. URL http://www.jstor.org/stable/2371320.
[16] S. Kumar, T.H. Lai, A. Arora, Barrier coverage with wireless sensors, in: MobiCom'05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, ACM, New York, NY, USA, 2005, pp. 284–298.
[17] T. Razafindralambo, D. Simplot-Ryl, Connectivity preservation and coverage schemes for wireless sensor networks, IEEE Trans. Automat. Control 56 (10) (2011) 2418–2428.
[18] Z. Wang, Formation control in mobile actuator/sensor networks, in: SPIE, no. 435, Spie, 2005, pp. 706–717.
[19] G. Wang, G. Cao, T.F.L. Porta, Movement-assisted sensor deployment, in: IEEE INFOCOM, 2004, pp. 2469–2479.
[20] B. Wang, H.B. Lim, D. Ma, A survey of movement strategies for improving network coverage in wireless sensor networks, Comput. Commun. 32 (13–14) (2009) 1427–1436.
[21] J. Whitbeck, M. Dias de Amorim, V. Conan, J.-L. Guillaume, Temporal reachability graphs, in: MOBICOM, 2012, pp. 377–388.
[22] WifiBot, Mobile robots plateform, www.wifibot.com (2016). URL www.wifibot.com.
[23] WSNet, An event-driven simulator for large scale wireless sensor networks, (2016). URL.
[24] M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: A survey, Ad Hoc Networks 6 (4) (2008) 621–655.
[25] Z. Yun, X. Bai, D. Xuan, T.H. Lai, W. Jia, Optimal deployment patterns for full coverage and k-connectivity ($k \leq 6$) wireless sensor networks, IEEE/ACM Trans. Netw. 18 (3) (2010) 934–947. URL http://dx.doi.org/10.1109/TNET.2010.2040191.
[26] D. Zorbas, T. Razafindralambo, Modeling the power consumption of a wifibot and studying the role of communication cost in operation time, Tech. rep, Inria Lille – Nord Europe, 2015, 11.
[27] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, Vol. 2, IEEE Societies, 2003, pp. 1293–1303.

Tahiry Razafindralambo received his M.Sc. in applied statistics and computer science from the university of Antananarivo in 2001 and his Ph.D. degree in Computer Science from the INSA de Lyon in 2007. He is currently an Inria full researcher. His research interests are mainly focused on distributed algorithms and protocols design for wireless networks and performance evaluation. He is involved in many organization and program committees of national and international conferences such as DCOSS, MASS, PE-WASUN, MSWIM, PIMRC, ICC and he is the principal investigator of many national and international projects.

Milan Erdelj received his Ph.D. from Inria Lille—Nord Europe where he was member of the FUN team. He is currently post-doctoral researcher at the Université de Technologie de Compiègne, in the Network & Optimization group within the Heudiasyc Lab.

Dimitrios Zorbas received his Ph.D. in 2011 from the University of Piraeus in Greece. During 2011 and 2014 he was member of the FUN team at Inria Lille—Nord Europe in France as a post doctorate researcher. He is currently researcher at University of La Rochelle and works on autonomous systems and distributed networks. He has, also, worked on several EU projects in Greece and in France.

Enrico Natalizio started obtained his Ph.D. in 2005 from the University of Calabria (Italy). From October 2010 to August 2012 he worked with the FUN Team at Inria Lille - Nord Europe as a postdoctoral researcher. In 2012, he joined the Network & Optimization group within the Heudiasyc Lab at the Université de Technologie de Compiègne as an Associate Professor. His main research activities focus on UAV, robot and sensor networks, swarm communications and wireless multimedia networks. He is Associate Editor of three international journals and has been Guest Editor of 7 Special Issues.